Martin Beckmann

Protokollierung von CAN-Nachrichten mithilfe eines integrierten Data Warehouse

Martin Beckmann (2014): Protokollierung von CAN-Nachrichten mithilfe eines integrierten Data Warehouse. In: Informatiktage 2014 - Big (Data) is beautiful. Bonn: Gesellschaft für Informatik.
URI: http://dl.gi.de/handle/20.500.12116/4930.

Protokollierung von CAN-Nachrichten mithilfe eines integrierten Data Warehouse

Martin Beckmann

TU Berlin
Fachgebiet Softwaretechnik, DCAITI
martin.beckmann@mailbox.tu-berlin.de

Art der Arbeit: Bachelorarbeit
Betreuer der Arbeit: Thomas Noack

Abstract: Diese Arbeit beschäftigt sich mit der Entwicklung eines Werkzeuges zur Verarbeitung von CAN-Nachrichten von Steuergeräten. Die Nachrichten werden anschließend protokolliert, um eine Auswertung im Rahmen von Softwaretests zu ermöglichen. Dazu wurde eine Hardwarelösung entwickelt und eine passende Software implementiert.

1 Motivation

Für die vielfältigen und komplexen Aufgaben in Automobilen werden zunehmend Steuergeräte zur Bewältigung dieser Aufgaben genutzt [Saa03]. Es ist notwendig, dass diese spezialisierten Steuergeräte dabei untereinander kommunizieren können. Als Beispiel sei genannt, dass die Lautstärke eines Entertainmentsystems in Abhängigkeit von der Geschwindigkeit des Fahrzeuges angepasst wird. Daran sind Komponenten der Fahrdynamik, der Nutzerinteraktion und des Infotainmentsystems beteiligt. Ermöglicht wird diese Kommunikation durch die Verwendung des CAN-Bus [Alb04]. Bei dem Controller Area Network (CAN) handelt es sich um ein serielles Bussystem, welches speziell auf die Anforderungen von Steuergeräten in Automobilen ausgelegt ist [11806].

Um Fehlfunktionen bei dieser Komplexität erkennen und beheben zu können, werden die Komponenten Integrationstests unterzogen. Bei diesen Tests werden voneinander abhängige Komponenten kombiniert und getestet. In diesem Zusammenhang geht es um die Kommunikation auf dem CAN-Bus zwischen den Steuergeräten in einem Verbund. Die Integrationstests sind unvermeidbar, um Qualitätseinbußen und auch Fehlfunktionen sicherheitsrelevanter Funktionen zu vermeiden [AS12]. Die Kommunikation der Steuergeräte ist dabei sowohl während der Integration in ein Teilsystem als auch in das Gesamtsystem von besonderer Bedeutung.

Der Einsatz von vielen Komponenten erzeugt ein hohes Datenaufkommen auf dem Bus. Um die Wechselwirkungen unter den Steuergeräten und deren Funktion kontrollieren zu können, ist eine Aufzeichnung der CAN-Nachrichten notwendig. Dies ist weiterhin erforderlich, da eine große Anzahl an relevanten Testfällen existiert und somit die Möglichkeit der späteren Diagnose besteht. Um das Verhalten reproduzierbar zu überprüfen, muss die Buskommunikation über einen längeren Zeitraum festgehalten werden. Dadurch wird im Nachhinein eine korrekte Testdurchführung nachweisbar.

Ausgehend von dieser Situation wurde eine Hardware entworfen und umgesetzt, welche an ein CAN-Netzwerk angeschlossen werden kann. Für diese Hardware sollte anschließend eine Software implementiert werden, die eine Überwachung im Betrieb ermöglicht, Daten dauerhaft speichert und einen Export zum späteren Zeitpunkt erlaubt. Die Lösung wurde im Rahmen einer Bachelorarbeit in Kooperation mit der *P3 systems GmbH* umgesetzt. Es folgt die Beschreibung der Implementierung.

2 Implementierung

Um eine Lösung zur Protokollierung in eine integrierte Datenbasis von CAN-Bus Nachrichten zu erhalten, ist es notwendig eine Kombination aus Hardware und Software zu verwenden. Die Architektur der Lösung ist in Abbildung 1 dargestellt.

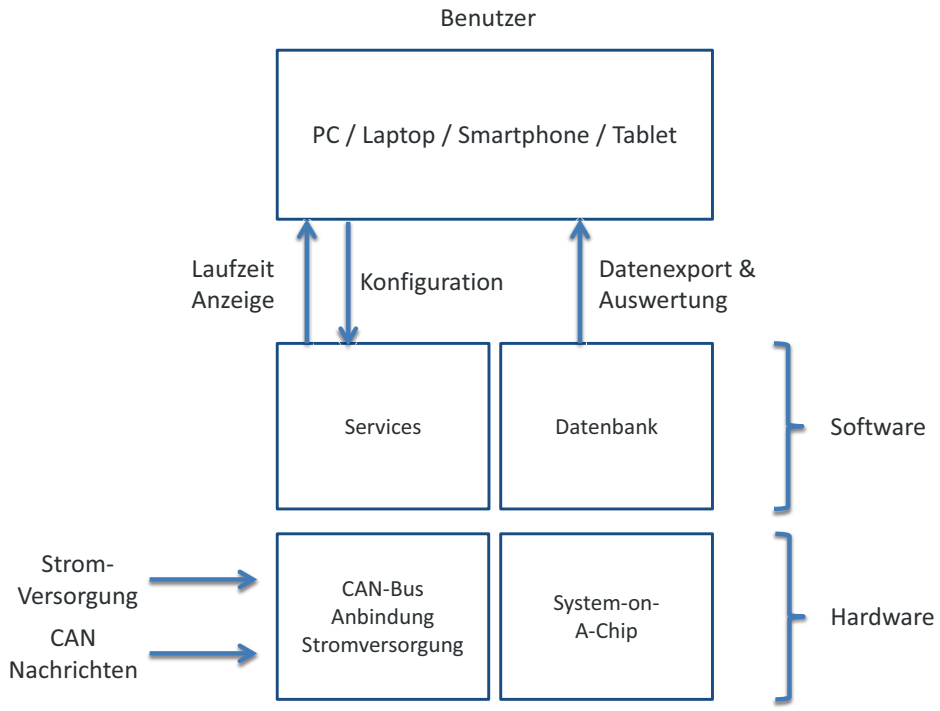

Abbildung 1: Architektur der Lösung

2.1 Hardware

Als Grundstein wird ein weitverbreitetes System-on-a-Chip (Raspberry Pi) verwendet [Fou14]. Wegen seiner Kompaktheit ist es flexibel sowohl im Labor als auch im Fahrzeug einsetzbar. Auf der Hardware wird ein Linux-Betriebssystem ausgeführt, welches u.a. dazu dient eine Datenbank zur Protokollierung zu betreiben.

Zur Anbindung an einen CAN-Bus wurde der Raspberry Pi um selbstentwickelte Hardware erweitert, welche aus zwei Platinen besteht. Diese Platinen werden aufgesteckt. Die eine Platine stellt hierbei die Stromversorgung bereit. Darauf aufgesteckt wird die zweite, ebenfalls in Eigenentwicklung entstandene Platine zur Verarbeitung der CAN-Nachrichten.

2.2 Software

Zum sinnvollen Betrieb der Hardware ist außerdem noch eine Reihe an Softwarebestandteilen entwickelt worden. Auf unterster Ebene ist dies ein Programm, welches die CAN-Nachrichten von der CAN-Hardware einliest. Diese CAN-Nachrichten werden anschließend in ein für Menschen lesbares Format umgewandelt und in ihre jeweiligen Bestandteile zerlegt. Das Zerlegen der Nachricht dient dazu selbige in einer Datenbank zu speichern.

Des Weiteren wurde eine graphische Benutzeroberfläche entwickelt. Darin eingebettet sind Funktionalitäten zur Konfiguration des Gerätes. Dies reicht von Einstellungen bezüglich des CAN-Busses zum Beispiel der Symbolrate bis zu Einstellungen der Datenorganisation.

Der Datenbestand, welcher zur weiteren Auswertung genutzt wird, ergänzt außerdem elementare Informationen wie Sitzungsdauer und die Summe der protokollierten Nachrichten zu gespeicherten Testläufen.

Die Datenbank selbst speichert nicht nur Informationen über die CAN-Nachrichten, wie Bezeichner, Daten der Nachricht, usw. sondern auch essentielle Informationen wie den Zeitstempel der Nachricht oder auf welchem CAN-Bus diese gesendet wurde.

Dabei werden wichtige Eigenschaften eines Data Warehouse erfüllt, die notwendig sind, um einen Softwaretest erfolgreich und belastbar durchführen zu können. Am wichtigsten sind dabei die nicht flüchtige Datenbasis, welche die Unveränderlichkeit der Daten garantiert sowie die historische Datenbasis, welche für Vergleiche herangezogen werden kann.

Neben der eigens entwickelten Software wurden auch bereits etablierte Werkzeuge eingesetzt. Sämtliche eingesetzte Software steht frei zu kommerziellen Nutzung zur Verfügung und konnte daher ohne Weiteres verwendet werden.

3 Fazit und Ausblick

Die erstellte Lösung ist flexibel, klein und mobil genug, um die Ansprüche, die im Labor und im realen Testfahrzeug vorliegen, erfüllen zu können. Neben der reinen Protokol-

lierung der Kommunikationsdaten, kann im Betrieb der Nachrichtenverlauf nachvollzogen werden. Aufgrund der Verwendung von etablierten Bauteilen und freien Werkzeugen, konnte dies kostengünstig realisiert werden.

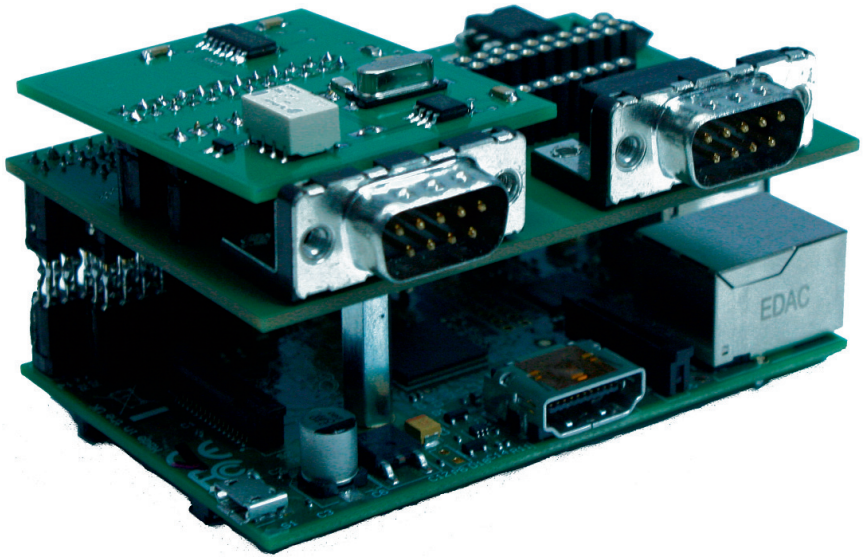

Abbildung 2: Komplett aufgebaute Hardware

Ein Einsatz der Lösung ist möglich ohne weitere Geräte zu nutzen. Deren Einsatz bietet jedoch zusätzliche Funktionalität. Die Art des Gerätes (Laptops, Smartphone, Tablet) ist hierbei frei wählbar, da die Lösung komplett plattformunabhängig arbeitet.

Eine Erweiterung der Arbeit ist sowohl auf der Hard- als auch der Softwareebene möglich. Da das Hardwarekonzept sehr flexibel gestaltet wurde, ist das Hinzufügen von Elektronik zum Protokollieren von anderen Bussystemen (z.B. FlexRay) oder von Messelektronik denkbar. Auf Seiten der Software sind als weitere Schritte die Dateninterpretation und Datenanalyse möglich.

Literatur

[11806] ISO 11898-1. *ISO 11898-1:2003 Road vehicles Controller area network Part 1: Data link layer and physical signalling*, 2006.

[Alb04] Amos Albert. Comparison of event-triggered and time-triggered concepts with regard to distributed control systems. *Embedded World*, 2004:235–252, 2004.

[AS12] Tilo Linz Andreas Spillner. *Basiswissen Softwaretest*. dpunkt.verlag GmbH, 5. Auflage, 2012.

[Fou14] The Raspberry Pi Foundation. A birthday present from Broadcom, 2014. Available online at http://www.raspberrypi.org/archives/6299; visited on March 1th 2014.

[Saa03] Alexandre Saad. Das Automobil als Anwendungsgebiet der Informatik-ein Auto ohne Informatik, geht das? In *INFOS*, Seiten 37–40, 2003.